



AF
Gfa

THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellant: GOERING, Thomas

Confirmation No. 5450

Appl'n No.: 10/665,305

Group Art Unit: 2176

Filing Date: 22 September 2003

Examiner: Rachna Singh

For: SYSTEM AND METHOD FOR REUSING FORM
ELEMENTS IN A FORM BUILDING APPLICATION

Mail Stop APPEAL BRIEF – PATENTS

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

AMENDED APPEAL BRIEF RESPONSE UNDER 37 C.F.R. 41.37

Dear Sir:

This amended brief is in furtherance of the Notice of Appeal filed on November 14, 2006.

In response to the Notification of Non-Compliant Appeal Brief mailed on February 20, 2008, for the above-identified application, submitted herewith include:

- **CORRECTIONS MADE TO SECTION “I. SUMMARY OF CLAIMED SUBJECT MATTER”,** which begins on page 2;
- **REMARKS** which begins on page 4; and
- **APPENDIX: AMENDED APPEAL BRIEF** which is attached.

CORRECTIONS MADE TO SECTION "I. SUMMARY OF CLAIMED SUBJECT MATTER"

Please note the following corrections to the section indicated. The underlined portion was added. There were no deletions.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The subject matter defined in the three independent claims on appeal (claims 1, 9 and 17) is directed generally to methods and systems for the generation of output modules in a form-based application runtime environment. The forms are created using a form building application. (Specification, ¶ 8).

The form-based application runtime environment is an application for reducing the amount of programming skills necessary for creating and maintaining forms and to enable users to design the look-and-feel of business forms in a graphical environment without coding. (Specification, ¶ 2).

The system and methods of claims 1, 9, and 17 enable form elements to be shared by multiple forms and for the changes to a form element to trigger the regeneration of the output module that use the changed form element in order to incorporate the changes. (Specification, ¶ 8). An output module is described as a function module that can subsequently be called by an application, for example, to create delivery notes in Sales and Distribution. The form output module processes the imported application-specific data and its form description data for presentation via spool (printer), fax, e-mail, web browser, etc. Since retrieval of application-specific data is performed by the application program and then passed to the form output module, a clean separation of the data retrieval processes is established from the form design processes so that only changes to the form layout or form logic are made in the form building application. (Specification, ¶ 6).

A form manager component is a program for managing reusable form elements by associating forms with reusable form elements. (Specification, ¶¶ 39, 40). Reusable form elements are form elements that can be shared among multiple output modules, e.g. the corporate address, or in general, parts of a form. (Specification, ¶¶ 7, 45). The runtime manager is a

program to regenerate invalidated form output modules before calling them. (Specification, ¶ 39). Note the fact that higher-level form elements cannot be merely incorporated by reference into form output modules, but rather must be generated into the output module. (Specification, ¶ 8).

More particularly, the system and methods of the present invention enable an output module to support a reusable form element that has been changed after being incorporated into a form, such that that form outputted by the output module reflects the most recent change made to the reusable form element.

FIG. 2 illustrates an embodiment of the invention as recited in independent claims 1, 9 and 17 (Specification, page 5 at lines 9-15) in which an indication is received that a reusable form element has been changed (step 210), a determination is made as to which output modules from a set of output modules are affected by the changed form element (step 220), and the affected output modules are invalidated (step 230).

The embodiment further illustrates that a request is received for an output module from the set of output modules (step 240), and the requested output module is regenerated (step 260) if the requested output module has been invalidated (“no” branch of step 250).

This embodiment is described in the specification at least in para. 29.

REMARKS

The appeal brief received on December 6, 2007 for the above referenced case is held non-compliant for failing to comply with 37 CFR. § 41.37(c)(1)(v).

The Notification mailed on February 20, 2008, states the brief does not map every limitation and the subject matter of each independent claim to the Specification. Particularly, Appellant should map the recited “output modules” to those portions of the Specification that describe these elements in detail. The corrections noted in **CORRECTIONS MADE TO SECTION “I. SUMMARY OF CLAIMED SUBJECT MATTER”** remedy the above noted deficiencies.

Applicant asserts the amended brief is now in compliance with 37 C.F.R. 41.37.

CONCLUSION

Therefore, all defects having been addressed, it is respectfully submitted that the amended appeal brief is in condition to be considered by the Board of Patent Appeals and Interferences. The Examiner is invited to contact the undersigned at (202) 220-4200 to discuss any matter concerning this application.

Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 11-0600 and please credit any excess fees to such deposit account.

Respectfully submitted,



Gregory R. Grace
Registration No. 59,733

Dated: March 20, 2008

KENYON & KENYON LLP
1500 K Street, N.W., Suite 700
Washington, D.C. 20005
(202) 220 - 4200 (telephone)
(202) 220 - 4201 (facsimile)

Serial No. 10/665,305
Amended Appeal Brief filed March 20, 2008



APPENDIX: AMENDED APPEAL BRIEF



**IN THE UNITED STATES PATENT AND
TRADEMARK OFFICE**

Appellant: GOERING, Thomas

Appl'n No.: 10/665,305

Group Art Unit: 2176

Filing Date: 22 September 2003

Examiner: Singh, Rachna

For: SYSTEM AND METHOD FOR REUSING FORM
ELEMENTS IN A FORM BUILDING APPLICATION

Mail Stop APPEAL BRIEF – PATENTS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDED APPEAL BRIEF UNDER 37 C.F.R. 41.37

SIR:

This brief is in furtherance of the Notice of Appeal filed on November 14, 2006, and in response to the Notification of Non-Compliant Appeal Brief mailed on February 20, 2008, for the above-identified application.

I. REAL PARTY IN INTEREST

SAP Aktiengesellschaft is the real party in interest for all issues related to this application.

II. RELATED APPEALS AND INTERFERENCES

None.

III. STATUS OF CLAIMS

Pending claims 1-18 stand finally rejected and are the subject of this appeal. Claim 19 is canceled.

IV. STATUS OF AMENDMENTS

Subsequent to the August 14, 2006 final Office action [hereinafter "Final Rejection"], the October 26, 2006 Amendment After Final (37 C.F.R. § 1.116) was entered for purposes of appeal in the November 29, 2006 Advisory Action.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The subject matter defined in the three independent claims on appeal (claims 1, 9 and 17) is directed generally to methods and systems for the generation of output modules in a form-based application runtime environment. The forms are created using a form building application. (Specification, ¶ 8).

The form-based application runtime environment is an application for reducing the amount of programming skills necessary for creating and maintaining forms and to enable users to design the look-and-feel of business forms in a graphical environment without coding. (Specification, ¶ 2).

The system and methods of claims 1, 9, and 17 enable form elements to be shared by multiple forms and for the changes to a form element to trigger the regeneration of the output modules that use the changed form element in order to incorporate the changes. (Specification, ¶ 8). An output module is described as a function module that can subsequently be called by an application, for example, to create delivery notes in Sales and Distribution. The form output

module processes the imported application-specific data and its form description data for presentation via spool (printer), fax, e-mail, web browser, etc. Since retrieval of application-specific data is performed by the application program and then passed to the form output module, a clean separation of the data retrieval processes is established from the form design processes so that only changes to the form layout or form logic are made in the form building application. (Specification, ¶ 6).

A form manager component is a program for managing reusable form elements by associating forms with reusable form elements (Specification, ¶¶ 39, 40). Reusable form elements are form elements that can be shared among multiple output modules, e.g. the corporate address, or in general, pieces of a form. (Specification, ¶¶ 7, 45). The runtime manager is a program to regenerate invalidated form output modules before calling them. (Specification, ¶ 39). Note the fact that higher-level form elements cannot be merely incorporated by reference into form output modules, but rather must be generated into the output module. (Specification, ¶ 8).

More particularly, the system and methods of the present invention enable an output module to support a reusable form element that has been changed after being incorporated into a form, such that that form outputted by the output module reflects the most recent change made to the reusable form element.

FIG. 2 illustrates an embodiment of the invention as recited in independent claims 1, 9 and 17 (Specification, page 5 at lines 9-15) in which an indication is received that a reusable form element has been changed (step 210), a determination is made as to which output modules from a set of output modules are affected by the changed form element (step 220), and the affected output modules are invalidated (step 230).

The embodiment further illustrates that a request is received for an output module from the set of output modules (step 240), and the requested output module is regenerated (step 260) if the requested output module has been invalidated (“no” branch of step 250).

This embodiment is described in the specification at least in para. 29.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

- whether claims 1-18 are unpatentable under 35 U.S.C. §103(a) over U.S. Patent Application Publication No. 2005/0080756 A1 to Hitchcock [hereinafter “Hitchcock”]

VII. ARGUMENT

The Final Rejection fails to demonstrate that Hitchcock renders obvious any of pending claims 1-18 for at least the reasons that Hitchcock does not teach or suggest:

- invalidating output modules as recited in independent claims 1, 9 and 17, and
- regenerating invalidated output modules as recited in independent claims 1, 9 and 17

Further, the Examiner’s position equating the claimed output module with mere form data, which is fundamental to the Examiner’s obviousness argument, constitutes clear error and should not be sustained.

Details of these arguments are presented below.

A. Claims 1-18 Are Not Rendered Obvious By Hitchcock

i) Hitchcock Fails To Teach or Suggest Invalidating Output Modules

Independent claim 9 recites, in part, “invalidating . . . output modules”. Independent claim 1 includes similar recitations. Independent claim 17 recites, in part, “determining whether a[n] output module . . . has been marked as invalid”.

No aspect of Hitchcock, taken alone or in combination, teaches or suggests invalidation of output modules as claimed.

In fact, Hitchcock explicitly teaches *against* invalidating the output modules by stating that “[t]he applicant database can be extended to include new attributes *without making any changes to the forms engine program*”. Hitchcock, para. 0065, lns 1-3 (emphasis added).

Because invalidation would lead to changes to the forms engine program, Hitchcock cannot be reasonably cited to teach or suggest the above-recited claim language.

(1) **The Examiner's Position Equating the Claimed Output Module with Form Data Constitutes Clear Error**

The Examiner's rejection of the invalidating output modules claim language is based squarely on the erroneous position that the claimed output module can be equated with form data.

Specifically, the Examiner stated:

It is the Examiner's view that the entry of new values in the pre-populated field invalidates the pre-populated fields throughout the other applications because the new values are now being used.

Final Rejection, p. 16, lns 7-9.

To illustrate the Examiner's position with an example, suppose a college applicant using Hitchcock's system pulls up a college admission application form that has a phone number field pre-populated with the applicant's phone number. If the applicant were to type in a new phone number in the phone number field, the Hitchcock system would store the new number so that it would be automatically inserted into the phone number fields of other college admission application forms subsequently pulled up by the applicant.

The Examiner's position is that the entry of mere form data (e.g., the text of a phone number inputted into a form) is equivalent to invalidating output modules as claimed.

It is clear from the specification and plain meaning that an "output module" is a module that provides output, and is not the output itself. In the context of forms, a form output module processes form data for presentation (e.g., via printer, web browser, etc.). See specification, p. 2, lns 14-16.

Thus, because form data is *clearly and factually distinct and separate from* a form output module, the Office's position reading form data on the claimed output module lacks factual basis, constitutes clear error and should not be sustained.

ii) **Hitchcock Fails To Teach or Suggest Regenerating Invalidated Output Modules**

Independent claim 9 recites, in part, “regenerating [a] requested output module if the requested output module has been invalidated”. Independent claim 1 includes similar recitations. Independent claim 17 recites, in part, “determining whether a[n] output module . . . has been marked as invalid” and “if so: regenerating the output module”.

No aspect of Hitchcock, taken alone or in combination, teaches or suggests regeneration of invalidated output modules as claimed.

Since the essence of Hitchcock is to provide an output module - or forms engine under Hitchcock's nomenclature - that is extensible *without programming* (see Hitchcock, Abstract), Hitchcock teaches *against* regenerating invalidated output modules and therefore cannot be reasonably cited to teach or suggest the above-recited claim language.

(1) **The Examiner's Position Equating the Claimed Output Module with Form Data Constitutes Clear Error**

The Examiner's rejection of the regenerating invalidated output modules claim language is also based squarely on the erroneous position that the claimed output module can be equated with form data.

Specifically, the Examiner stated:

Hitchcock teaches new forms are automatically populated with the previously entered data.

Final Rejection, p. 17, lns 2-3.

Thus, in accordance with the above college admission application form example, the Examiner's position is that the insertion into a form of mere form data (e.g., the text of a phone number inputted into a form) is equivalent to regenerating invalidated output modules as claimed.

However, as stated above, because form data is *clearly and factually distinct and separate from* a form output module, the Office's position reading form data on the claimed output module lacks factual basis, constitutes clear error and should not be sustained.

iii) **The Dependent Claims Are Likewise Not Rendered Obvious**

Regarding claims 2-8, 10-16 and 18, these claims depend from independent claims 1, 9 and 17 respectively, which, as detailed above, are not rendered obvious by Hitchcock. For at least the reason that the secondary arguments provided by the Examiner do not remedy the above-noted deficiencies of Hitchcock, these claims cannot be deemed obvious.

VIII. CONCLUSION

Applicant respectfully requests that the Board of Patent Appeals and Interferences reverse the Examiner's decision rejecting claims 1-18 and direct the Examiner to pass the case to issue. These claims are allowable over the cited art.

Respectfully submitted,

Dated: March 20, 2008



Gregory R. Grace
(Registration No. 59,733)

KENYON & KENYON LLP
1500 K Street, N.W., Suite 700
Washington, D.C. 20005
(202) 220 - 4200 (telephone)
(202) 220 - 4201 (facsimile)

CLAIMS APPENDIX

1. A computer system for generating output modules in a form-based application runtime environment, comprising:
 - a form manager component configured to receive an indication that a reusable form element has been changed, determine which output modules from a set of output modules are affected by the changed form element, and invalidate the affected output modules; and
 - a runtime manager component configured to receive a request for an output module from the set of output modules and cause regeneration of the requested output module if the requested output module has been invalidated.
2. The system of claim 1, wherein the indication is received when changes to the reusable form element are saved.
3. The system of claim 1, wherein the affected output modules are determined by referencing a record data structure.
4. The system of claim 1, wherein the affected output modules are invalidated by marking a flag associated with each affected output module as invalid.
5. The system of claim 1, wherein the request for the output module received by the runtime manager is a request to identify the output module.

6. The system of claim 1, wherein the reusable form element is one of a form page and a form window.

7. The system of claim 1, wherein the reusable form element is form logic.

8. The system of claim 1, wherein the reusable form element is a form interface.

9. A computer-implemented method for generating output modules in a form-based application runtime environment, comprising:

receiving an indication that a reusable form element has been changed;

determining which output modules from a set of output modules are affected by the changed form element;

invalidating the affected output modules;

receiving a request for an output module from the set of output modules; and

regenerating the requested output module if the requested output module has been invalidated.

10. The method of claim 9, wherein the indication is received when changes to the reusable form element are saved.

11. The method of claim 9, wherein the affected output modules are determined by referencing a record data structure.

12. The method of claim 9, wherein the affected output modules are invalidated by marking a flag associated with each affected output module as invalid.

13. The method of claim 9, wherein the request for the output module received by the runtime manager is a request to identify the output module.

14. The method of claim 9, wherein the reusable form element is one of a form page and a form window.

15. The method of claim 9, wherein the reusable form element is form logic.

16. The method of claim 9, wherein the reusable form element is a form interface.

17. A computer-implemented dynamic form building method, comprising:
responsive to a call to start a form output process based on an identified form:
determining whether a previously generated output module associated with the identified form in an output module library has been marked as invalid;
if so:

regenerating the output module; and

storing the regenerated output module in the output module library along with a marker to indicate that the output module is valid.

18. The method of claim 17, wherein the regeneration of the output module includes compiling changed reusable form elements into the output module.

EVIDENCE APPENDIX

Exhibit A

U.S. Patent Application Publication No. 2005/0080756 A1 to Hitchcock

- cited by the Examiner in the March 6, 2006 Office action and relied upon as to grounds of rejection to be reviewed on appeal

RELATED PROCEEDINGS APPENDIX

None.